IN THE SPECIFICATION

Please replace the paragraph beginning at page 14, line 25, with the following amended version of the paragraph.

> --Fig. 4 is a diagram illustrating how higher levels of parallelism may reduce the idle time, or waste, of processor resources. Included in fig. 4 are three different ways a given processor may issue instructions to its functional units. A superscalar 402, multithreaded superscalar 404, and simultaneous multithreaded superscalar 406 are shown. The processor illustrated includes three functional units which are represented by the issue slots 410A-410C, 412A-412C, and 414A-414C. Nine processor clock cycles 420-428 are represented in nine rows of fig. 4 with time increasing from earlier 420 to later 428.--

Please replace the paragraph beginning at page 16, line 20, with the following amended version of the paragraph.

> --Fig. 5 is a flowchart illustrating one embodiment of the load prediction and thread creation mechanism. In block 502, instruction window or buffer is scanned for load instructions. If a load is detected (decision block 504), flow continues to decision block 506. If no load is detected in block 504, control remains with block 502. In decision block 506, a load prediction table is searched for an entry which corresponds to the detected load instruction. If no entry is found for the detected load instruction, execution continues without a prediction, blocks 508 and 510. Subsequent to executing the unpredicted load, an entry is created (block 512) for the load in the load prediction table. On the other hand, if an entry for the detected load is found in the load prediction table (block 506), the effective address of the load is

calculated (block 518) and a miss count indicator in the table is checked (block 520) to determine if a load miss is predicted. If a load miss is indicated (block 520), a determination is made as to whether a thread slot is available (block 524). If no thread slot is available, an additional thread is not setup. On the other hand, if a thread slot is available, the load prediction unit scans (block 528) for the first instruction of a new thread (block 530). In one embodiment, when the first instruction of a new thread is found (block 530), information regarding the new thread is conveyed to the dispatch unit (block 532). Such information may include the address of the first instruction of the new thread and a thread unit identifier. Also, subsequent to computing the effective address (block 518) of a detected load, the predicted load is issued (block 522) and executed (block 526). If the predicted load subsequently hits in the data cache (block 534), an indication of this fact along with related information is conveyed to the load prediction unit where the corresponding load prediction table entry is updated (block 538). In one embodiment, this table entry update includes entering the difference between the previous effective address and the current effective address in a stride field of the corresponding entry. In addition, the update includes entering the actual effective address in the table entry. On the other hand, if a cache miss occurs (block 534) a fetch of the data is required (block 536) and an indication of this miss is conveyed to the load prediction unit. The corresponding load prediction table entry is then updated as before (block 538), with the addition of incrementing a miss counter (block 540).--

Please replace the paragraph beginning at page 19, line 27, with the following amended version of the paragraph.

--Now turning to fig. 7, a block diagram of one embodiment of a dispatch unit 106, two thread units 110A-110B and three functional units 140A-140C are shown. Dispatch unit 106 is coupled to load prediction unit via bus 330 and to thread units 110 via buses 750A and 750B. Thread units 110 are coupled to bus 180 which is also coupled to functional units 140. Thread units 110A-110B include, as shown, an instruction address register 710A-710B, instruction queue 712A-712B, instruction reordering and dependency checking circuitry 770A-770B, status registers 790A-790B, and decode units 720A-720D. In addition, thread unit 110B includes a first PC register 711. Instruction queues 712 are coupled to decode units 720. Circuitry 770 is coupled to instruction queue 712 and decode units 720. Status register 790 includes a reservation bit and an active bit. Also, in one embodiment, one thread unit 110A may be considered the main thread unit. The main thread unit 110A executes all single threaded code and may be the source for additional threads of execution.--

Please replace the paragraph beginning at page 14, line 25, with the following amended version of the paragraph.

-- Finally, instruction sequence 606 illustrates a third way of selecting a first instruction in a new thread. Sequence ~~604~~ 606 includes 47 instructions of which instructions 1, 2, 26, 46 and 47 are shown. In sequence 606, instructions 2 through 46 may represent the body of an iterative loop. Instruction 1 is a LD of register R3 with an initial value. Subsequently, in instruction 26, the value of R3 is decrement by a decrement instruction, DECR. Finally, instruction 46 represents a test of the value of R3. If the value of R3 is greater than zero, the control returns to instruction 2. Otherwise, control passes to instruction 47. In this instruction sequence, instruction 47, the instruction immediately following a loop iteration branch instruction, is selected as the first instruction in a new thread. Consequently, the address of the instruction and the received thread unit ID are conveyed to the dispatch unit where a new thread may be initialized and executed.--